REMARKS/ARGUMENTS

Claims 1-20 are pending in the present application. Reconsideration of the claims is respectfully requested.

I. Interview Summary

Applicant thanks the Examiner for the courtesies extended during the interview that was held on January 11, 2007. Applicant's claims and the cited prior art were discussed. No agreement was reached during the interview.

II. 35 U.S.C. § 103, Obviousness

A. Claims 1-5, 7-13 and 15-20 over *Arndt* in view of *Stine*

The Examiner has rejected claims 1-5, 7-13 and 15-20 under 35 U.S.C. § 103(a) as being unpatentable over *Arndt*, Logical Partitioning Via Hypervisor Mediated Address Translation, U.S. Patent No. 6,877,158 (April 5, 2005) (hereinafter "*Arndt*") in view of *Stine* et al., Memory Allocation System, U.S. Patent No. 6,629,111 (September 30, 2003) (hereinafter "*Stine*"). This rejection is respectfully traversed.

Applicant's independent claims recite similar features. Applicant's claim 1 recites "virtualizing a memory mapped input/output physical address range that is intermediate the first physical address range and the second physical address range to produce a third logical address range, wherein a lowermost logical address of the third logical address range exceeds a respective uppermost logical address of the first and second logical address ranges."

The Examiner stated that *Arndt* does not teach "virtualizing a memory mapped input/output physical address range that is intermediate the first physical address range and the second physical address range to produce a third logical address range, wherein a lowermost logical address of the third logical address range exceeds a respective uppermost logical address of the first and second logical address ranges" and relies on *Stine* to supply this feature that is missing from *Arndt*. Applicant respectfully disagrees that *Stine* teaches this feature.

Stine teaches a method of allocating memory in systems that restrict the mapping between physical addresses and virtual addresses to those existing on page boundaries. Virtual memory is allocated to a library when the library is to be accessed. This virtual memory is allocated in pages. Since a library typically does not occupy exactly a page, there is a portion of the page that is unused. Thus, a "hole" exists in a virtual memory page when a library, which is loaded into a virtual page, is not large enough to occupy the entire virtual page. Similarly, there is a corresponding "hole" in physical memory page that was mapped to the virtual page.

Stine takes advantage of the empty virtual memory storage that is left when a library does not occupy the entire virtual page that was allocated for that library. When another library needs to be loaded into virtual memory, *Stine* searches for an available "hole" in the allocated virtual memory. If a hole is found that is the same size or larger than the library that needs to be loaded, the library is loaded into the hole in virtual memory. Thus, in *Stine*, two different libraries can be loaded into the same page. See *Stine*, column 10, lines 30-51.

The new library is addressed using a virtual address that is calculated by adding together the virtual address of the original library already loaded into the page plus the size of the original library. Thus, the virtual address that is used to address the new library,had already been allocated to the existing library that was already loaded into the page. In this manner, no new virtual addresses need to be allocated when the new library is loaded into memory. *Stine* makes this point clear utilizing Figure 10 and its related description in column 10, lines 9-29.

Stine does not teach "virtualizing a memory mapped input/output physical address range that is intermediate the first physical address range and the second physical address range to produce a third logical address range, wherein a lowermost logical address of the third logical address range exceeds a respective uppermost logical address of the first and second logical address ranges".

Stine does not teach virtualizing physical memory of one of these "holes". *Stine* searches for a hole in the allocated virtual address space. If a hole is found, that hole is in a virtual memory page that was already mapped to physical memory. Therefore, nothing is virtualized. The virtualization of physical memory to virtual memory had already been completed. *Stine* teaches merely a reassignment of virtual addresses from an existing library to a new library.

Stine does not teach a third logical address range. In *Stine*, the logical address range of the hole, into which the new library is loaded, is not a third logical address range. In fact, the logical address range of the hole, into which the new library is loaded, is not a new logical address range at all. The starting virtual address for the new library is the starting virtual address of the hole. This is made clear in *Stine* in Figure 10. *Stine*, column 10, lines 23-26, which states: "More particularly, the starting address of the library is obtained by adding the starting virtual address of the previously loaded library to the size of the previously loaded library."

Stine does not teach "wherein a lowermost logical address of the third logical address range exceeds a respective uppermost logical address of the first and second logical address ranges". As discussed above, existing logical addresses are used for the new library that is loaded into the hole.

Therefore, there is no lowermost logical address of the third logical address range because there is no third logical address range.

Stine does teach allocating a new virtual address range when an existing hole does not exist. When a new library needs to be loaded and there is not an existing hole that is large enough into which the new library can be loaded, a new virtual page must be allocated. This new virtual page is allocated starting at the next available address in the virtual memory. See *Stine*, column 10, lines 2-5.

While *Stine* does teach allocating the virtual memory at the next available address, *Stine* does not teach the next available address having a lowermost logical address that exceeds a respective uppermost logical address of the first and second logical address ranges. The next available virtual address could have an address that is lower than the other virtual addresses. Nothing in *Stine* teaches the next available address exceeding the virtual addresses in which the other libraries are loaded.

Stine does not teach virtualizing a memory mapped input/output physical address range that is intermediate the first physical address range and the second physical address range to produce a third logical address range, wherein a lowermost logical address of the third logical address range exceeds a respective uppermost logical address of the first and second logical address ranges. Because neither *Arndt* nor *Stine* teaches virtualizing a memory mapped input/output physical address range that is intermediate the first physical address range and the second physical address range to produce a third logical address range, wherein a lowermost logical address of the third logical address range exceeds a respective uppermost logical address of the first and second logical address ranges, the combination of *Arndt* and *Stine* does not render Applicant's claims obvious.

The Examiner relies on *Arndt* to teach virtualizing a first physical address range allocated for system memory for an operating system run by a processor configured to support logical partitioning to produce a first logical address range; and virtualizing a second physical address range allocated for system memory for the operating system to produce a second logical address range, wherein the first physical address range and the second physical address range are non-contiguous and the first logical address range and the second logical address range are contiguous. Applicant respectfully disagrees that *Arndt* teaches these features.

Applicant claims virtualizing a first physical address range allocated for <u>system memory for an operating system</u> and virtualizing a second physical address range allocated for <u>system memory for the operating system</u>. *Arndt* does not teach virtualizing physical addresses that are allocated for system memory for an operating system. *Arndt* teaches physical hardware addresses that correspond to a <u>physical resource</u>.

The first and second physical address ranges claimed by Applicant are operating system memories, see Applicant's Figure 2, OS memory 210 and OS memory 211. The physical addresses in

Arndt have not been allocated for system memory for an operating system. They are not operating system memories. They have been allocated for physical resources, such as I/O devices.

Neither *Arndt* nor *Stine* teaches virtualizing a first physical address range allocated for system memory for an operating system run by a processor configured to support logical partitioning to produce a first logical address range; and virtualizing a second physical address range allocated for system memory for the operating system to produce a second logical address range, wherein the first physical address range and the second physical address range are non-contiguous and the first logical address range and the second logical address range are contiguous. Therefore, the combination of *Arndt* and *Stine* does not render Applicant's claims obvious.

Neither *Arndt* nor *Stine* teaches the features of Applicant's claims. Therefore, the rejection of claims 1-5, 7-13 and 15-20 under 35 U.S.C. § 103(a) has been overcome.

B. Claims 6 and 14 over *Arndt* in view of *Stine*, and further in view of *Yazdy*

The Examiner has rejected claims 6 and 14 under 35 U.S.C. § 103(a) as being unpatentable over *Arndt* in view of *Stine* as applied to claims 1-5 above, and further in view of *Yazdy* et al., Cache Management During Cache Inhibited Transactions for Increasing Cache Efficiency, U.S. Patent No. 6,256,710 (July 3, 2001) (hereinafter "*Yazdy*"). This rejection is respectfully traversed.

Applicant's claim 6 recites: "wherein the memory mapped input/output physical address range is allocated for cache inhibited addresses".

Applicant's claim 14 recites: "wherein the third range of contiguous physical addresses is allocated for cache inhibited memory mapped input/output addresses".

The Examiner states that the combination of *Arndt* and *Stine* does not teach the physical addresses being allocated for cache inhibited memory mapped input/output addresses. The Examiner relies on *Yazdy* to teach this feature.

Yazdy teaches an area of main memory that is non-cacheable. See *Yazdy*, column 2, lines 23-14. *Yazdy* does not, however, teach physical addresses that are allocated for cache inhibited memory mapped input/output addresses. The area of main memory that is non-cacheable is not described as being allocated for cache inhibited memory mapped input/output addresses.

None of the cited references teaches physical addresses that are allocated for cache inhibited memory mapped input/output addresses; therefore, the combination of *Arndt*, *Stine,* and *Yazdy* does not teach physical addresses that are allocated for cache inhibited memory mapped input/output addresses. Therefore, the combination of *Arndt*, *Stine,* and *Yazdy* does not render claims 6 and 14 obvious.

The rejection of claims 6 and 14 under 35 U.S.C. § 103(a) has been overcome.

III. Conclusion

It is respectfully urged that the subject application is patentable over *Arndt, Stine* and *Yazdy*, and is in condition for allowance.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: January 19, 2007

Respectfully submitted,

/Lisa L.B. Yociss/

Lisa L.B. Yociss
Reg. No. 36,975
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicant